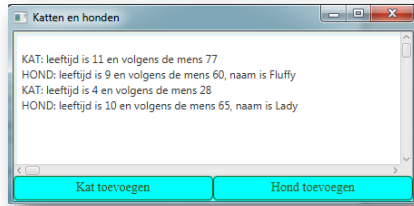


# Oefening

De gebruiker kan een kat of een hond toevoegen.

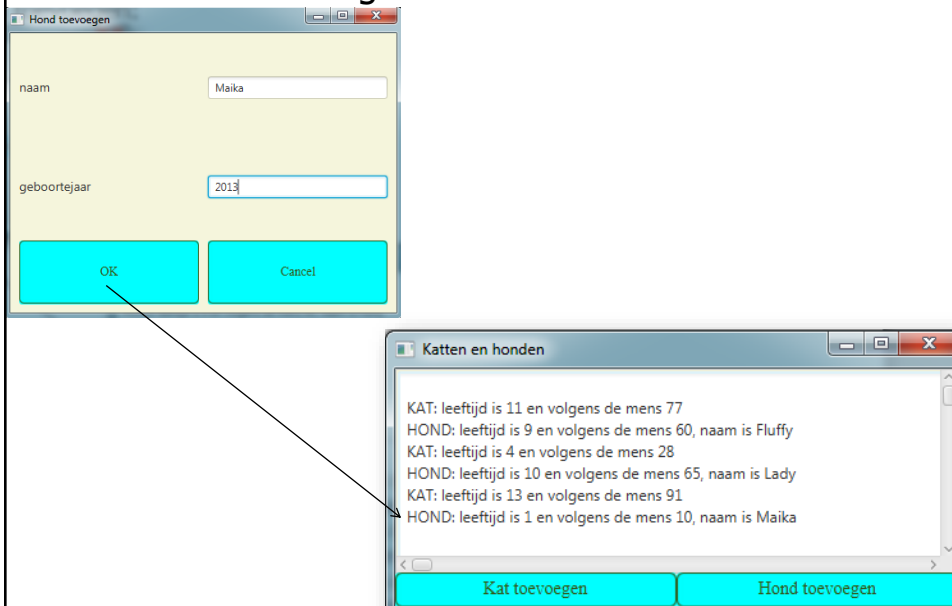


Vb. een kat toevoegen:

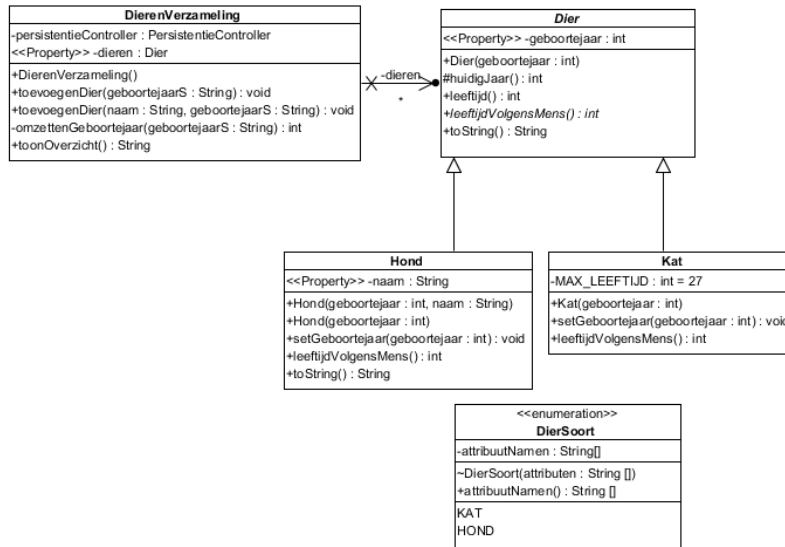


# Oefening

Vb. een hond toevoegen:

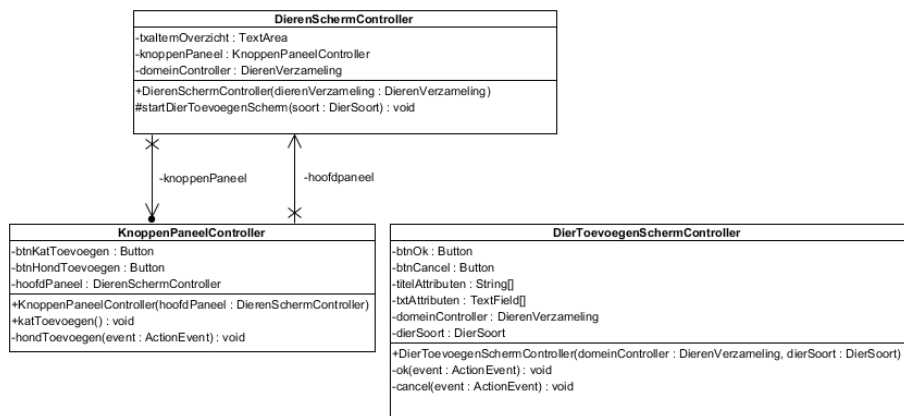


## package domein:



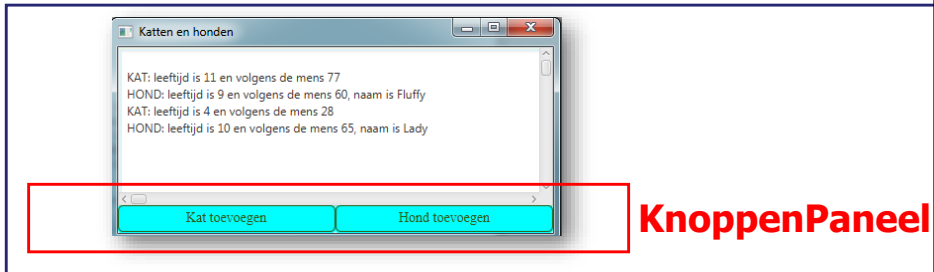
3

## package gui (oplossing 1 TIN):



## package gui

### DierenScher



### subscher

### DierToevoegenScher



package gui (oplossing 1 TIN):

KnoppenPanneel

```
public class KnoppenPaneelController extends GridPane {
```

```
    @FXML
```

```
    private Button btnKatToevoegen;
```

```
    @FXML
```

```
    private Button btnHondToevoegen;
```

```
    private final DierenScherController hoofdPanneel;
```

```
    public KnoppenPaneelController(  
        DierenScherController hoofdPanneel) {
```

```
        this.hoofdPanneel = hoofdPanneel;
```

```
        ...
```

```
    }
```

```
package gui (oplossing 1 TIN):
```

```
KnoppenPaneel
```

```
@FXML
```

```
private void katToevoegen(ActionEvent event)
```

```
{
```

```
    hoofdPaneel.
```

```
        startDierToevoegenScherM(DierSoort.KAT);
```

```
}
```

```
@FXML
```

```
private void hondToevoegen(ActionEvent event)
```

```
{
```

```
    hoofdPaneel.startDierToevoegenScherM(
```

```
        DierSoort.HOND);
```

```
}
```

```
package gui (oplossing 1 TIN):
```

```
DierenScherM:
```

```
public DierenScherMController(DierenVerzameling dierenVerzameling) {
```

```
    this.domeinController = dierenVerzameling;
```

```
    knoppenPaneel = new KnoppenPaneelController(this);
```

```
    ...
```

```
}
```

```
protected void startDierToevoegenScherM(DierSoort soort) {
```

```
    Stage stage = new Stage();
```

```
    stage.setTitle(soort.equals(DierSoort.KAT) ?
```

```
        "Kat toevoegen" : "Hond toevoegen");
```

```
    stage.setScene(new Scene(
```

```
        new DierToevoegenScherMController(domeinController, soort));
```

```
    this.setDisable(true);
```

## package gui (oplossing 2 TIN):

### Pas het Observer Pattern toe

